

VARIABLE LOCALE ȘI GLOBALE → C++

Aspecte teoretice generale:

Orice variabilă C++ se caracterizează prin atributele:

1. **Clasa de memorare;**
2. **Vizibilitatea;**
3. **Durata de viață;**
4. **Identificatorul variabilei;**
5. **Tipul variabilei;**
6. **Zonă de memorie.**

1. CLASA DE MEMORARE

Clasa de memorare precizează locul unde este memorată variabila. Spațiul rezervat, de sistemul de operare, unei variabile, poate fi:

- într-un segment de date, spațiu rezervat variabilelor globale;
- în stivă, pentru parametrii funcțiilor și variabile locale
- în zona heap, variabilele alocate dinamic.

2. VIZIBILITATEA

Vizibilitatea precizează liniile textului sursă din care variabila poate fi accesată. Se întâlnesc următoarele situații:

- **Vizibilitate la nivel de bloc:** variabila este recunoscută într-un bloc din momentul declarării până la sfârșitul blocului.
- **Vizibilitatea la nivel de fișier:** variabila este recunoscută într-un fișier; în cazul în care programul ocupa un fișier sursă.
- **Vizibilitatea la nivel de clasă:** va fi tratată la programarea orientate pe obiecte.

3. DURATA DE VIAȚA

Durata de viață precizează timpul în care variabila respectivă are alocat spațiu în memoria internă. Se întâlnesc următoarele situații:

- **Durata statică:** variabila are alocat spațiu în memoria internă în tot timpul execuției programului.
- **Durata locală:** variabila are alocat spațiu în memoria internă în timpul în care se execută instrucțiunile blocului în care a fost declarată.
- **Durata dinamică:** variabila are alocat spațiu, în **heap**, în funcție de necesitățile programatorului, prin funcții sau operatori specifici.

4. IDENTIFICATORUL VARIABILEI

Identificatorul este numele dat variabilei în momentul declarării sale, cu scopul identificării între variabile diferite. În general, primele 31 de caractere sunt semnificative.

5. TIPUL VARIABILEI

Tipul variabilei precizează apartenența variabilei la un domeniu de valori înzestrat cu operații specifice și numărul de octeți rezervați pentru memorarea datei respective.

6. ZONA DE MEMORIE

Fiecare variabilă are alocată o zonă proprie în memorie, precizată prin adresa de început și dimensiune (număr de octeți).

În funcție de atributele pe care le au, variabilele se clasifică astfel:

- variabile globale;
- variabile locale;
- variabile dinamice.

VARIABLE GLOBALE

Variabilele globale se caracterizează astfel:

- declarația definiție** a lor este **în afara oricărei funcții**;
- li se alocă spațiu **într-un segment de date**, înainte de execuția programului;
- au durata de **viață statică**;
- sunt inițializate implicit cu 0; pot fi inițializate explicit numai cu valori constante;
- vizibilitatea lor poate fi extinsă la nivelul întregului program.

Acestea se declară în afara corpului oricărei funcții, ca mai jos:

```
#include<iostream>
```

```
int a,
```

```
int t()
```

```
{ a=3;
```

```
  cout<<a;
```

```
}
```

```
int b;
```

```
main()
```

```
{ b=4;
```

```
  cout<<a<<endl;
```

```
t();
```

```
}
```

Variabilele a și b sunt globale. În astfel de cazuri, variabilele respective pot fi utilizate de toate funcțiile *care urmează în textul sursă declarației variabilei respective*. Din acest motiv, astfel de variabile se numesc **globale**.

!!! La declarare, variabilele globale sunt **inițializate implicit cu 0**.

VARIABLE LOCALE

Variabilele locale se clasifică astfel:

- variabile locale automate;
- variabile locale statice.

Variabilele locale automate sunt

- declarația definiție a lor se face **în interiorul unui bloc**, și vizibilitatea este la nivelul acestuia;
- li se alocă spațiu pe stivă sau în registrele microprocesorului – atunci când declarația definiție a lor este precedată de specificatorul **register**, când se întâlnește declarația definiție;
- sunt **dezalocate** când se iese din blocul respectiv;
- pot fi **inițializate explicit** cu valori constante sau cu valori variabile;
- **vizibilitatea lor nu poate fi extinsă** în afara blocului în care au fost declarate.

Acestea sunt declarate în corpul funcțiilor. Mai precis, pot fi declarate în orice bloc (instrucțiune compusă) al acestora.

!!!Variabilele declarate în corpul funcției main() **sunt tot locale**. În programul următor, variabilele a și b sunt locale.

Variabila a este declarată în corpul funcției t(), iar variabila b este declarată în corpul funcției main().

Exemplu:

```
void t()
{ int a=3;
}
```

```
main()
{ int b=4;
}
```

***Exemplu :** în care apar variabile locale automate:

```
#include <iostream>
void f()
{
  int i=1,j=i;
  cout<<i<<" "<<j<<endl;
  {
    int i=2,j=i;
    cout<<" "<<j<<" "<<endl;
    {
      int i=3,j=i;
      cout<<" "<<i<<" "<<j<<endl;
    }
    cout<<i<<" "<<j<<endl;
  }
  cout<<i<<" "<<j<<endl;
}
```

```
main()
{
  f();
}
```

Se afișează:

```
1 1
  2 2
    3 3
  2 2
1 1
```

Comentariu: Exemplul anterior ilustrează faptul că domeniul de vizibilitate al unei variabile locale automate este blocul în care se găsește declarația definiție a acesteia.

Observație: În același bloc nu pot fi declarate mai multe variabile automate cu același nume, dar pot fi declarate variabile locale (automate) cu același nume în blocuri diferite!